

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

2013

Lukáš Šípinka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student:

Lukáš Šípinka

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R059 Mobilní technologie

Téma:

**Absolvování individuální odborné praxe
Individual Professional Practice in the Company**

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ITA spol. s.r.o. Ostrava
2. Struktura závěrečné zprávy:
 - a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
 - b. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
 - c. Zvolený postup řešení zadaných úkolů
 - d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
 - e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
 - f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

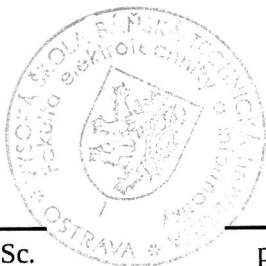
Vedoucí bakalářské práce: **Ing. Zdeňka Chmelíková, Ph.D.**


Konzultant bakalářské práce: **Ing. Pavel Šimeček, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013


prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
2.5.2013

V Ostravě dne

.....
Šípinka

L. Šípinka

Poděkování

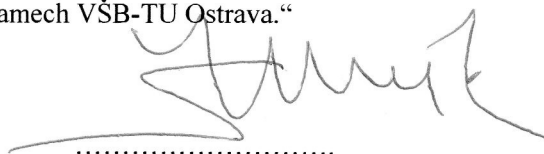
Rád bych poděkoval vedoucí bakalářské práce Ing. Zdeňce Chmelíkové, Ph.D. za odbornou pomoc a konzultaci při vytváření této bakalářské práce. Zároveň bych chtěl poděkovat všem kolegům z firmy ITA spol. s.r.o., zvláště pak Ing. Pavlu Šimečkovi, Ph.D. za skvělou podporu co se týče ochoty a přístupu při vykonávání zadaných úkolů.

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.“

2.5.2013

V Ostravě dne

A handwritten signature in black ink, appearing to read 'Šimeček', written over a dotted line.

Ing. Pavel Šimeček, Ph.D.

Abstrakt

Bakalářská práce si klade za cíl popsat moje pracovní nasazení ve firmě ITA spol. s.r.o. v rámci absolvování individuální odborné praxe. Ve firmě mi byly přiděleny různé úkoly, které jsem podle náročnosti vykonával s různou časovou prodlevou. Všechny zadané úkoly byly napsány v programovacím jazyce C++. Prvním úkolem, který jsem dostal, bylo vytvořit aplikaci pro třídění, která bude umět třídit vybrané skupiny kovů na základě jejich chemického složení. Aplikace byla zpočátku postavena na algoritmu, který mi byl přidělen. Ovšem během mé práce i na jiných aplikacích, byl tento algoritmus postupem času upravován a to pro jeho lepší funkčnost. Mým dalším úkolem bylo vytvořit aplikaci pro export dokumentů z formátu HTML do formátu PDF. Dále jsem se podílel na úpravě stávající firemní knihovny MetaCool, která poskytuje pestrou škálu výpočetních postupů a modulů v oblasti kovového průmyslu. Posléze jsem tuto knihovnu používal při tvorbě GUI aplikace, která by za účelem této knihovny zobrazovala 2D grafy.

Klíčová slova

ITA, odborná praxe, různé úkoly, C++, aplikace pro třídění, export dokumentů, úprava firemní knihovny, tvorba GUI aplikace

Abstract

Bachelor dissertation describes my work commitment in the company ITA Comp. s.r.o. in pursuance of attend to individual professional practice. In the company were assigned various tasks to me that I performed with different time delay according to the difficulty. All assignments were written in the programming language C++. The first task, which was given to me, was to create an application for sorting, which should be able to classify the selected group of metals on the basis of their chemical composition. At first, the application was based on an algorithm that was assigned to me. However, during my work also on other applications, this algorithm was modified in time, in order to have better functionality. My next task was to create an application for exporting of documents from HTML to PDF. Then I participated in the adjustment of the existing library MetaCool, that provides a wide scale of computational methods and modules in the field of metal industry. Then I used this library to create GUI application that would help the library display 2D graphs.

Key words

ITA, professional practice, various tasks, C++, application for sorting, export documents, adjustment of the existing library, creating GUI application

Seznam použitých symbolů, zkratek a termínů

GUI	Graphical User Interface
HTML	Hyper Text Markup Language
PDF	Portable Document Format
MDI	Multi Document Interface
SDI	Single Document Interface
MFC	Microsoft Foundation Class Library
CSS	Cascade Style Sheets
XML	Extensible Markup Language
API	Application Programming Interface
GphCtrls32	Graphic Controls 32
TableCtrl	Table Controls
SW	SoftWare

Obsah

1 Úvod.....	10
2 Popis firmy ITA a pracovního zařazení.....	11
2.1 ITA.....	11
2.2 Popis pracovního zařazení.....	11
3 Úkoly zadané během odborné praxe.....	12
3.1 Projekt Třídění ocelí do skupin.....	12
3.2 Projekt HtmlToPdf – tvorba Reportů.....	13
3.3 Úprava firemní knihovny MetaCool.....	13
3.4 Projekt SmallQT.....	14
4 Technologie.....	15
4.1 MFC.....	15
4.1.1 Výhody používání C++ s knihovnou MFC.....	15
4.2 GphCtls32.....	16
4.3 TableCtrl.....	17
4.4 LibHaru.....	17
4.5 MetaCool.....	18
5 Zvolený postup řešení zadaných úkolů.....	19
5.1 Třídění ocelí do skupin.....	19
5.2 HtmlToPdf.....	20
5.3 SmallQT.....	21
6 Znalosti a dovednosti získané během studia a uplatněné v průběhu odborné praxe.....	22
7 Znalosti a dovednosti scházející studentovi v průběhu odborné praxe.....	23
8 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.....	24
9 Seznam použité literatury.....	25
A Příloha.....	26

1 Úvod

Svou odbornou praxi jsem vykonával ve firmě ITA s. r. o., přičemž hlavním důvodem pro výběr této praxe byla v první řadě možnost podílet se na vývoji reálného softwaru, dále možnost týmové spolupráce, programování v jazyce C++ a v neposlední řadě také získání nových znalostí, zkušeností a firemního *know-how*. Během praxe mi byly přiděleny různé úkoly (např. projekt HtmlToPdf pro export Reportů) a zároveň jsem získal během jejich plnění spoustu nových znalostí. Rovněž jsem si mohl ověřit své schopnosti, které jsem nabyl během studia. Celý průběh odborné praxe, včetně popisu řešených úkolů, je součástí této práce. Zprv je ve stručnosti popsáno odborné zaměření firmy, moje pracovní zařazení a použité technologie. Dále se věnuji úkolům zadaným v průběhu odborné praxe, přičemž ke každému úkolu jsou uvedeny také postupy a technologie, které jsem pro jejich řešení použil. Ke konci jsou shrnuty znalosti a dovednosti, které jsem nabyl během absolvování odborné praxe. A nakonec, v závěru této bakalářské práce, jsou shrnuty moje dosažené výsledky a celkové zhodnocení odborné praxe.

2 Popis firmy ITA a pracovního zařazení

2.1 ITA

Společnost ITA je soukromá česká společnost založená v roce 1991 výzkumnými a vědeckými pracovníky Výzkumného ústavu strojírenského a metalurgického VÍTKOVICE. Sídli v Ostravě v městské části ulice Mariánské Hory. Zabývá se moderními technologiemi válcování za tepla i za studena. Dodává *know-how* a programová řešení významným dodavatelům válcovacích zařízení, technologií a řídicích systémů, jako např. Danieli, ConverTeam, Acos Vilares, Vítkovice Heavy Machinery. Řeší technické problémy a technologické inovace na teplých a studených válcovacích tratích, jako např. ArcelorMittal Ostrava, Severstal Čerepovec, ArcelorMittal Vanderbijlpark, CSN Voltaredunda, Třinecké železářny Třinec [1].

Mezi hlavní činnosti firmy patří např. vývoj nových, optimalizace a úpravy stávajících SW modulů řídicích systémů válcovacích tratí. Další činností firmy je jak vývoj speciálních programů určených pro *off-line* simulace procesů válcování a ochlazování, tak i konzultační, poradenská a další činnosti v oblasti technologií válcování za tepla i za studena.

2.2 Popis pracovního zařazení

Ve firmě ITA jsem byl zařazen na pozici programátora se zaměřením na programovací jazyk C++. Zpočátku jsem byl seznámen s pracovním prostředím a také s kolegy programátory, kteří mi byli neustále k dispozici při řešení jakéhokoliv problému. Po celou dobu vývoje mi byli po ruce jak kolegové, tak i především můj přímý nadřízený. Mým přímým nadřízeným byl pan Ing. Šimeček Ph.D., který mi zadával úkoly, analyzoval je se mnou a rovněž vymýšlel nové algoritmy (např. u aplikace pro třídění skupin ocelí) [8].

V průběhu celé praxe mi byly přiděleny různé úkoly, přičemž mým prvním úkolem bylo vytvořit GUI aplikaci, která by uměla třídit vybrané skupiny ocelí na základě jejich chemického složení. Tato aplikace byla založena na algoritmu, který byl, během vývoje na jiných aplikacích, postupně vylepšován (a mnou naprogramován) panem Šimečkem. Mým druhým úkolem bylo vytvořit vestavěnou aplikaci pro export webových stránek do formátu PDF, která by byla později přidána do již stávající aplikace. Třetím úkolem bylo upravit stávající firemní knihovnu MetaCool podle předem stanovených kritérií. Poté mým čtvrtým úkolem bylo vytvořit GUI aplikaci, která by otestovala funkčnost upravené knihovny (zobrazení grafů).

3 Úkoly zadané během odborné praxe

Před samotným plněním úkolů jsem se musel seznámit s technologiemi, které jsem v průběhu praxe používal. Všechny úkoly byly napsány v programovacím jazyce C++, přičemž technologie, které jsem pro své programovací účely využíval, byly např. knihovny MFC, GphCtls32, TableCtrl, LibHaru a další. S knihovnou GphCtls32 jsem byl seznámen ještě před samotným vykonáváním odborné praxe. To bylo dáno z důvodu většího počtu uchazečů o místo ve firmě. Byl nám proto přidělen úkol, jehož hlavním cílem bylo otestování grafické firemní knihovny GphCtls32. Úkol jsem splnil, a proto jsem byl také přijat na odbornou praxi ve firmě ITA [9].

3.1 Projekt Třídění ocelí do skupin

Na počátku mojí praxe jsem dostal za úkol naprogramovat MFC aplikaci, která by dovedla, na základě chemického složení oceli, určit, do které skupiny ocel patří nebo ke kterým skupinám má nejblíže. Jinak řečeno, aplikace dovede na základě chemického složení oceli třídit vybrané skupiny ocelí v rámci přiděleného algoritmu. Skupina, která je v žebříčku třídění na prvním místě, má nejblíže k zadanému chemickému složení. Aplikace byla zpočátku postavena na jediném algoritmu, ale postupem vývoje na jiných aplikacích byl algoritmus upravován a optimalizován mým nadřízeným.

Firma ITA používá pro své účely mimo jiné knihovny MFC a TableCtrl, a z toho důvodu jsem je používal neustále po celou dobu mojí praxe. Pro mou první aplikaci jsem používal obě knihovny. MFC knihovnu jsem v tomto případě používal pro vytvoření Windows dialogu, na kterém bych mohl zobrazit výsledky třídění. Výsledky třídění byly reprezentovány formou tabulky získanou z knihovny TableCtrl. Knihovna TableCtrl (včetně dokumentace) mi byla poskytnuta firmou. Celá MFC aplikace byla tedy reprezentována Windows dialogem, ve kterém byly umístěny kontrolní řídicí prvky spolu s tabulkou. Dále mi byl předán seznam skupin ocelí, kde každá skupina představovala jiné chemické složení. Tento seznam jsem znázornil v tabulce. Uživatel si pak jednoduše zadal chemické složení, a poté, kliknutím na tlačítko Setřídít, se tabulka setřídila. Tabulku bylo možné upravovat, přičemž výsledky se ukládaly do textového souboru. Třídění probíhalo na základě dvou úrovní pomocí dvou tlačítek, kde každá úroveň představovala jiný stupeň třídění (viz příloha, obr. č. 2 na straně 26).

3.2 Projekt HtmlToPdf – tvorba Reportů

Mým druhým úkolem bylo vytvořit aplikaci pro export Reportů ve formátu HTML do formátu PDF. Firma poskytuje svým klientům v rámci aplikací také tzv. reporty, které reprezentují informace o výstupech z jejich aplikací např. obrázky, různé zprávy o mechanických vlastnostech oceli, zprávy o kalení oceli, ochlazovací křivky, atd. Ty jsou zpřístupněny ve formátu HTML, ale pro potřeby svých klientů si firma přála, aby tyto zprávy mohly být k dispozici i ve formátu PDF.

Aplikaci jsem vytvořil s použitím knihovny LibHaru a to tak, aby se mohla přidat do již stávající firemní aplikace. Na konci tohoto dokumentu jsou k nahlédnutí obrázky, které znázorňují, jak vypadá dokument před samotnou konverzí a po konverzi (viz příloha, obr. č. 4 na straně 28 a obr. č. 5 na straně 29) [2].

3.3 Úprava firemní knihovny MetaCool

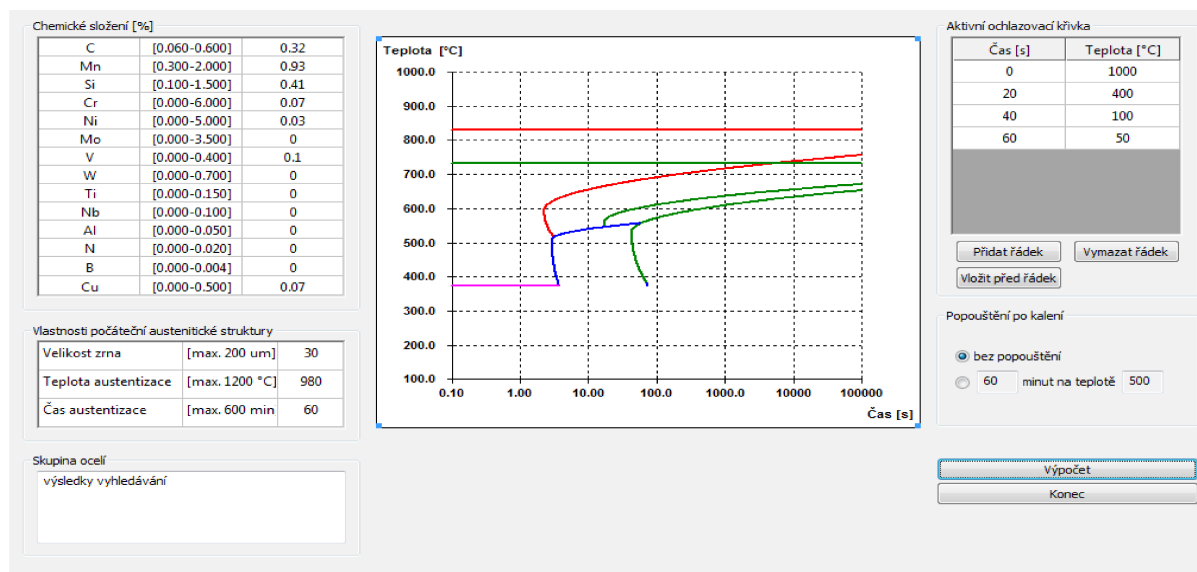
Mým třetím úkolem bylo upravit stávající knihovnu, která nese název MetaCool. Knihovna MetaCool obsahuje řadu funkcí a její kořenovou funkcí je funkce MetaCool. Všechny knihovní funkce slouží pro výpočty mechanických vlastností různých skupin ocelí. Knihovna tedy obsahuje funkce pro různé skupiny ocelí dané svým chemickým složením. Mým úkolem bylo přidat nové skupiny oceli a to tak, aby byly charakteristické a především identické s již známými skupinami ocelí. Ačkoli jsou stejné, tak hlavním důvodem bylo, aby firma mohla pro nové skupiny oceli vytvořit nové algoritmy, které by později mohla nasadit a testovat na oněch skupinách. Dále jsem na projektu MetaCool z části upravoval také její kořenovou funkci se stejnojmenným názvem MetaCool. Funkce slouží pro získání skupiny oceli. Ve výchozím nastavení měla funkce staticky nastavenou skupinu, ale mým úkolem bylo, abych do ní přidal algoritmus třídění, který jsem vytvářel v podobě méj první aplikace – Třídění ocelí do skupin. Funkce by pak vracela hodnotu reprezentující vybranou skupinu oceli podle chemického složení - výběr skupiny, ke které má chemické složení nejbližší.

Bylo pro mě docela náročné, než jsem pochopil chování této knihovny. Nebylo sice zapotřebí znát celé její chování, ale z důvodu dalšího vývoje na budoucí aplikaci SmallQT bylo potřeba chápat alespoň některé její podstatné složky. Později jsem musel knihovnu upravovat znovu, protože z předcházející úpravy vyšlo najevo, že obsahovala nějaké chyby. Nicméně tím hlavním důvodem byla nově zavedená koncepce, podle které by vznikla nová a efektivnější verze knihovny MetaCool (MetaCool 3.0).

3.4 Projekt SmallQT

Mým čtvrtým úkolem bylo vytvořit MFC aplikaci, která by na základě chemického složení oceli a jejích dalších parametrů zobrazovala grafy, které by reprezentovaly mechanické vlastnosti vybrané skupiny oceli. V projektu SmallQT jsem z velké části využíval knihovny TableCtrl a GphCtrls32, ale samozřejmě také řadu jiných knihoven. Knihovnu TableCtrl jsem použil pro vytvoření tabulek, např. tabulku pro nastavení chemického složení oceli a jiných tabulek co se týče zadávání uživatelských vstupních parametrů v aplikaci. Firemní knihovnu GphCtrls32 jsem naopak použil pro vytvoření grafů. Celá aplikace měla být propojena s firemní knihovnou MetaCool, kterou jsem upravoval a díky jejíž kořenové funkci MetaCool by se zjistilo, ke které skupině oceli má zadané chemické složení nejbližší.

Aplikaci jsem vytvořil v podobě Windows dialogu, se kterým se lépe zachází. Ovšem jak jsem posléze zjistil, bylo by lepší vybrat jako typ MFC architektury tzv. Dokument/Pohled [6]. A to z toho důvodu, jelikož architektura typu Dokument/Pohled má oproti architektuře typu Dialog mnohem lepší vlastnosti a schopnosti, které architektura typu Dialog postrádá. Běh aplikace je svázán s počátečním výpočtem a zobrazením grafu, který je umístěný ve středu aplikace. Poté má uživatel možnost použít tlačítko pro výpočet, po němž proběhne další výpočet včetně zavolání funkce MetaCool. Postup a další informace o aplikaci SmallQT jsou stručně shrnuty v kapitole - Zvolený postup řešení zadaných úkolů. Aplikace je k nahlédnutí níže (viz obr. č. 1), nebo také na konci tohoto dokumentu (viz příloha, obr. č. 3 na straně 27).



Obrázek č. 1: Dialogové okno SmallQT aplikace

4 Technologie

Jak jsem se již zmínil, musel jsem se kvůli zadaným úkolům seznámit s různými technologiemi, které se odvíjely podle toho, na čem jsem zrovna pracoval. Nyní je ve stručnosti představím.

4.1 MFC

MFC je knihovnou tříd jazyka C++, poskytovanou společností Microsoft k vytvoření objektově orientovaného obalu okolo rozhraní Windows API. Verze 6 obsahuje přibližně 200 tříd, z nichž některé se používají přímo, a další slouží jako základ pro vlastní definované třídy. Některé z obsažených tříd jsou neobvykle snadné (např. třída *CPoint*) zatímco ostatní třídy jsou poměrně složitější. K nim lze přiřadit kupříkladu třídu *CWnd*, jenž zapouzdřuje funkčnost okna.

MFC je také aplikačním rámcem. Je ne jenom kolekcí tříd, ale napomáhá také při definování struktury celé aplikace a vypořádá se s mnoha běžnými činnostmi ve prospěch dané aplikace. Jednou z nejdůležitějších tříd je třída *CWinApp*. Tato třída představuje samotnou aplikaci a MFC virtuálně zapouzdřuje všechny aspekty činnosti programu. Tento rámec podporuje funkci *WinMain*, která volá členské funkce objektu aplikace, a tím také zajišťuje chod programu. Jedna z členských funkcí třídy *CWinApp* volaná funkcí *WinMain*, funkce *Run*, poskytuje cyklus zpráv, jenž vyhledává zprávy určené oknu aplikace. Výhodou aplikačního rámce je také abstrakce, která zdaleka přesahuje rámec poskytovaný rozhraním Windows API. Například architektura MFC typu Dokument/Pohled vytváří výkonnou infrastrukturu nad rozhraním API. Odděluje data programu od jejich grafické reprezentace neboli od pohledu na tato data. Takové pojetí abstrakce je pro API zcela cizí a neexistuje mimo rámec MFC nebo jí podobné knihovny tříd. Více se o MFC knihovně dočtete zde [6].

4.1.1 Výhody používání C++ s knihovnou MFC

Nejlepší argument pro používání C++ s knihovnou MFC je bezpochyby ve prospěch metodologie objektově orientovaných návrhů, které dávají programům možnost opětovného používání, těsnější vazby kódu s daty atd. Při objektově orientovaném přístupu jsou pojmy jako objekt, dědičnost, polymorfismus nebo zapouzdření na předním místě, obzvláště pak patří-li k jazyku C++ [5]. Ve spojení s knihovnou MFC nám taková vazba nabízí širokou škálu použití. Ať už potřebujeme použít například panel nástrojů (třída *CToolBar*) nebo zřetězený seznam (třída *CList*) či

dynamické pole (třída *CArray*), tak nám MFC nabízí všechny tyto třídy včetně mnoha dalších, které pak dělají takřka většinu práce za nás. MFC toho nabízí samozřejmě mnohem více, ale to bohužel přesahuje rámec této bakalářské práce.

4.2 GphCtls32

GphCtls32 je podobně jako MFC knihovnou tříd jazyka C++, přičemž zdaleka nepřesahuje knihovnu MFC. Jedná se o grafickou knihovnu, která byla vytvořena pro pracovní účely firmy ITA, a která se zabývá vykreslováním 2D nebo 3D grafů. Nabízí širokou škálu tříd, které mají různé vlastnosti. Například třída *CLabel* slouží k vytvoření štítku, na kterém lze později umístit graf. Bez ní graf nelze umístit, pokud chceme použít MFC architekturu typu Dokument/Pohled. Je to tzv. „obálka grafů“, do níž je potřeba grafy zaobalit, aby mohly být zobrazeny. Dále například třída *CLegend* slouží k vytvoření legendy, která svým způsobem popisuje celý graf. Při tvorbě grafů je zapotřebí nejprve zvážit, chceme-li použít MFC architekturu typu Dialog anebo architekturu typu Dokument/Pohled. Architektura typu Dokument/Pohled je složitější a rozsáhlejší než architektura typu Dialog. Oproti tomu je architektura typu Dialog jednoduchá a pro skromné účely asi nejpoužívanější.

Všechny třídy knihovny GphCtls32 poskytují spoustu funkcí a metod, které jsou charakteristické pro mnoho účelů svými různými definicemi. Samotné třídy, které se používají pro práci s grafy, jsou třída *C2DGraphSet* a třída *C2DGraphSubSet*. Tyto třídy jsou dvě hlavní třídy pro práci s 2D grafikou. Data grafu musí být před jeho zobrazením inicializována. Inicializace se provádí prostřednictvím struktury, která se naplní. Obsahuje obecné vlastnosti grafu, jako např. dynamické pole bodů, název, atd. V rámci třídy *C2DGraphSet* lze umístit i více grafů, které nazýváme podmnožiny (třída *C2DGraphSubset*). Ve stručnosti je knihovna GphCtls32 knihovnou, která umí vytvořit 2D nebo 3D grafy, obsluhovat události, poskytovat informace o grafech a je na nás, jak si ji přizpůsobíme, což ve svém důsledku značí polymorfni chování [9].

4.3 TableCtrl

TableCtrl je knihovna, která poskytuje třídy pro tvorbu řádků a sloupců formou tabulky. Tabulku lze použít buď jako samostatné okno v MDI, v SDI, nebo také jako Dialog. Obsahuje celou škálu tříd, přičemž hlavní třídou je třída *CTableCtrlEx* nebo *CGridCtrl*, které reprezentují tabulku. Práce s knihovnou je velice jednoduchá, protože veškeré chování tabulky je zapouzdřeno. Pro tvorbu tabulky nám postačí mít vytvořenou proměnnou třídy *CTableCtrlEx* a volat patřičné metody např. pro nastavení počtu řádků, sloupců, atd.

Knihovna umí nejen zobrazovat obsah v tabulce, ale disponuje také celou řadou funkcí. Například dovede umístit do buňky, namísto pouhého textu, tlačítko třídy *ChoiceGroup* nebo jiné kontrolní prvky. Během vývoje na této knihovně si vývojáři dali asi největší práci při tvorbě kořenové tabulkové struktury. To znamená, že tabulka poskytuje rovněž kořenovou hierarchii, jakou mají např. XML dokumenty. Tabulka poskytuje také obsluhu zpráv, vyvolaných prostřednictvím uživatele, a mnoho dalších dovedností.

4.4 LibHaru

Během mojí práce na projektu HtmlToPdf neboli export dokumentů z formátu HTML do formátu PDF jsem použil knihovnu LibHaru [2]. LibHaru je multi-platformní, open-source knihovna sloužící pro generování PDF souborů v rámci aplikací napsaných v jazyce C nebo C++. Díky tomu, že knihovna slouží pro generování PDF souborů, disponuje celou řadou funkcí přes vykreslování čar, textu, obrázků až po nastavení fontu písma, zabezpečení, anotace textu a další. Podporuje také různé znakové sady včetně *UNICODE*, která byla pro mě prioritní, protože firma má své klienty i v zahraničí a pro správné zobrazení písma příslušné kultury je použití této znakové sady klíčové.

LibHaru obsahuje kořenové třídy, kterými jsou *HPDF_Doc* a *HPDF_Page*, kde první třída reprezentuje celý PDF dokument a druhá třída reprezentuje každou jeho stranu. Jak jsem již zmínil, obsahuje knihovna pestrou škálu funkcí, jakými jsou dále např. funkce pro nastavení fontu písma, funkce pro nastavení pozice textu, funkce pro získání výšky a šířky dokumentu a mnoho dalších. Pro psaní v dokumentu je typickou funkcí funkce *HPDF_Page_BeginText*, která signalizuje počátek pro textový zápis. Funkce *HPDF_Page_EndText* pak představuje analogii k oné funkci. Dále funkce *HPDF_Page_ShowText* slouží už pro samotné zobrazení textu [3].

4.5 MetaCool

MetaCool je podobně jako GphCtrls32 knihovnou, která byla vytvořena pro pracovní účely firmou ITA. Metodologie této knihovny je úzce svázána s různými skupinami ocelí, kde každá skupina poskytuje jiné výpočty mechanických vlastností oceli. Obsahuje spoustu funkcí, přičemž její kořenová funkce nese název *MetaCool*. Více se o knihovně můžete dočíst v části nazvané Úprava firemní knihovny MetaCool.

5 Zvolený postup řešení zadaných úkolů

Jelikož jsem během praxe dostával různé úkoly, které se mezi sebou vzájemně lišily, tak byla převážná většina aplikací postavena na bázi knihovny MFC. S touto knihovnou se mi poměrně dobře pracovalo, ikdyž na počátku mé praxe jsem byl kolikrát zmatený tím, jak to vlastně celé funguje. Bylo to pro mě zkrátka něco nového a velmi zajímavého. Nyní se ve stručnosti budu zabývat tím, jaké jsem zvolil postupy při řešení zadaných úkolů.

5.1 Třídění ocelí do skupin

U této aplikace jsme zpočátku přemýšleli nad tím, jaký algoritmus by se nejlépe hodil pro samotné třídění. Třídění probíhalo na skupinách, které byly charakteristické svým vlastním chemickým složením. Veškerá chemická složení jednotlivých skupin byla ukládána do formy textového souboru, který se načítal vždy po spuštění aplikace. Každé složení se skládalo z prvků, kde každý prvek měl určitý rozsah - minimum a maximum. Z těchto rozsahů jsme vycházeli pro zavedení prvního algoritmu třídění (viz příloha, obr. č. 2 na straně 26).

První algoritmus spočíval v myšlence, při níž se provedl výpočet u každé skupiny oceli pro získání kritéria D, který představoval tzv. stupeň tolerance. Čím vyšší bylo kritérium D, tím vyšší byla pravděpodobnost, že se zadané chemické složení hodí k dané skupině. Poté jsme zavedli druhý algoritmus, který fungoval v zásadě podobným způsobem, jenom k němu přibyly nějaké změny, jako např. metoda lineární interpolace [10]. Pomocí metody lineární interpolace jsme mohli získávat lepší odhady pro takové případy, kdy se konkrétní prvek ze zadaného chemického složení nenacházel v příslušném intervalu. Pro tyto účely jsme vytvořili proměnnou, jejíž hodnota představovala hloubku interpolační linie mezi body A a B. Tyto body byly odvozeny z příslušného intervalu daného chemického prvku. Mezi těmito body pak existovalo nějaké spojení neboli přímka, kterou nazýváme lineární interpolací. Pokud se hodnota ze zadaného složení nacházela mimo tyto body, pak s pomocí interpolace se určila hodnota „hloubky interpolace“, která se potom přiřadila k aktuální hodnotě kritéria D. Z toho jsme tedy vycházeli a to bylo také víceméně důvodem, proč se nám druhý algoritmus osvědčil jako nejlepší. Veškeré algoritmy, v rámci této aplikace, byly koncipovány mnou a především mým přímým nadřízeným panem Ing. Šimečkem, Ph.D [8].

5.2 HtmlToPdf

Na počátku projektu HtmlToPdf jsem se snažil najít nějakou knihovnu, která by dokázala provést přímou konverzi z formátu HTML do formátu PDF v rámci programovacího jazyka C++. To se mi bohužel nepodařilo. Nicméně jsem po nějaké době našel knihovnu, pomocí níž bylo možné vytvářet PDF dokumenty v rámci C++. Knihovna se nazývá LibHaru [2].

Jelikož jsem neměl k dispozici knihovnu pro přímou konverzi, tak jsem byl z toho důvodu nucen vytvořit si svůj vlastní HTML *parser*. *Parser* pracoval na jednoduché technice *parsování* - získávání textového obsahu z HTML dokumentů. Technika *parsování* byla založena na principu sledování tagů neboli značek. *Parser* jako takový měl ve svém jádře nadefinovány své vlastní značky, které byl schopný rozpoznat. Ovšem před samotnou definicí konkrétních značek bylo nutné, abych si zajistil, jaké značky firma používá ve svých reportech. To bylo dáno především z toho důvodu, protože jazyk HTML je obrovským značkovacím jazykem, který v sobě skrývá nemalou škálu příkazů pro formátování hypertextových dokumentů. Proto bylo nutné, abych se obrátil pouze na ty značky, které byly pro firemní reporty identické. Proces *parsování* potom fungoval v členské metodě *parseru*, která se krok po kroku procházela celým dokumentem, a sledovala ty části dokumentu, ve které se vyskytovaly ty konkrétní značky, pro které byl *parser* nastavený. Z takto „vysledovaných“ značek jsem mohl potom získat např. samotný text, vlastnosti písma (velikost, barva, atd.). Dále také např. identifikátory objektů pro aplikaci kaskádových stylů a mnoho dalších důležitých informací. Jakmile byl dokument úspěšně *zparsován*, tak ještě muselo proběhnout načtení kaskádových stylů. Každý dokument obsahoval ve své hlavičce také kaskádové styly, s jejichž přispěním se veškerý webový obsah snadněji rozvíjí ať už po stránce grafické, ale i obsahové [11].

Konečnou fází exportu HTML stránky pak již představuje samotná tvorba PDF dokumentu s použitím knihovny LibHaru. To se provádí prostřednictvím oněch *zparsovaných* informací. Při práci na této aplikaci jsem narazil na někdy drobné a leckdy poněkud větší problémy, nicméně tím největším problémem byla pro mě tvorba dynamické tabulky. Knihovna LibHaru totiž neobsahuje přímé funkce pro tvorbu tabulek, ale pouze např. funkce pro vykreslování geometrických tvarů, jako např. přímek, obdélníků atd. S využitím těchto a jiných vlastností knihovny se mi to nakonec podařilo. Tabulka se chová dynamicky, což ve výsledku znamená, že se umí vizuálně přizpůsobit podle délky textu (viz příloha, obr. č. 4 na straně 28 a obr. č. 5 na straně 29) [2].

5.3 SmallQT

Díky práci na projektu SmallQT jsem získal snad nejvíce znalostí a zkušeností. Bylo to dáno především z toho důvodu, že se v tomto projektu objevovaly věci, se kterými jsem se již na praxi setkal, ale ještě jsem se s nimi nestačil pořádně seznámit - probrat je do větší hloubky [4]. Například firemní knihovna GphCtrls32, tabulková knihovna TableCtrl a především knihovna MFC [6].

S využitím služeb knihovny MFC jsem si vytvořil dialogové okno rozšířením třídy *CDialog*, které reprezentovalo celou aplikaci. Bohužel, jak jsem v průběhu praxe zjistil, bylo by lepší použít MFC architekturu typu Dokumentu/Pohled, protože pro účel této aplikace byla architektura tohoto typu přímo ideální. Výběrem cesty dialogového okna jsem si svou práci bohužel trochu přiosťřil, ale právě díky tomu jsem si mohl vyzkoušet, jaké to je, když jsem v blízkosti kritických situací. Díky tomu jsem nabyl mnoho nových poznatků. Chybami se člověk zkrátka učí a já tedy postavil aplikaci na bázi dialogu. Po vytvoření dialogu rozšířením třídy *CDialog* jsem do dialogu postupně přidával různé prvky aplikace.

S použitím tříd *C2DGraphWnd* a *C2DGraphSet* z knihovny GphCtrls32 jsem do dialogu mohl přidávat grafy. Veškeré vykreslování grafů pak probíhalo v rámci knihovny GphCtrls32 a výpočty pak s pomocí knihovny MetaCool. Samozřejmě jsem využíval i řadu dalších knihoven. Při vykreslování podmnožin grafů bylo zapotřebí použít lineární interpolaci, se kterou jsem se ve firmě poměrně často setkával [10]. Ta umožnila odhadnout souřadnice těch bodů, které spojovaly nějakou přímkou danou dvěma body. Největším problémem, se kterým jsem se setkával, byl tzv. *memory leak* neboli únik paměti. Ale právě to byl důvod, který mě velmi ponaučil v otázkách pro správu paměti.

6 Znalosti a dovednosti získané během studia a uplatněné v průběhu odborné praxe

Veškeré zkušenosti a znalosti, které jsem na praxi využíval, byly nejvíce nabyty v předmětech Algoritmy I a Algoritmy II, kde jsem získal základní znalosti z hlediska programování. Nejcennější znalosti jsem získal zejména v předmětu Algoritmy II, protože hlavní náplní tohoto předmětu byl právě programovací jazyk C++ [5][7]. Díky studiu jsem také získal zkušenosti v rámci HTML a CSS, které jsem využíval zejména u aplikace pro export firemních reportů z formátu HTML do formátu PDF [11]. Také díky předmětům Programovací jazyky II a Architektura .NET, ve kterých se probíral především programovací jazyk C Sharp, jsem získal lepší pohled a zkušenosti při objektově orientovaném návrhu.

Převážná část materiálů, ať už to byly dokumentace či učebnice získané ve firmě, v knihovně nebo na internetu, byly v angličtině. Tudíž jsem si mohl ověřit své dosavadní jazykové znalosti, kterých jsem nabyl a které jsem mohl působením ve firmě nadále rozvíjet.

7 Znalosti a dovednosti scházející studentovi v průběhu odborné praxe

V průběhu moji odborné praxe mi scházela celá řada znalostí týkajících se převážně technologií, se kterými jsem se ve firmě průběžně seznamoval. Nicméně díky vstupnímu pohovoru, který nám uchazečům ve výsledku určil, abychom samostatně vypracovali projekt, jsem nabyl spoustu znalostí, ze kterých jsem mohl ve firmě nadále čerpat. Onen „vstupní projekt“ jsme dostali zadaný za účelem ověření znalostí především v rámci firemní knihovny GphCtls32 spolu s knihovnou MFC [9][4][6]. Jak již jsem předestřel, byly pro mě knihovny MFC a GphCtls32 mými prvními úskalími, protože jsem se s nimi nikdy předtím nesetkal.

Dalším úskalím byl pro mě samotný programovací jazyk C++. V průběhu praxe jsem se totiž setkával s takovými typickými problémy, jakými jsou kupříkladu práce s ukazateli, správa paměti, přetypování, práce s používanými zdroji (např. MFC Resources), standardizace a výčet dalších problémů, které mě posunuly zase o krok kupředu [5][7]. Naštěstí díky veliké ochotě mého nadřízeného včetně kolegů programátorů jsem se s problémy dokázal vypořádat. Rovněž i díky skvělé dokumentaci, která mi byla poskytnuta.

8 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Jsem velice rád, že jsem využil možnosti absolvování bakalářské práce formou podnikové praxe. Odborná praxe mi celkově přiblížila fungování firmy ITA zabývající se reálným vývojem softwaru. Získal jsem možnost pracovat v kolegiálním prostředí s lidmi, od kterých jsem se toho spoustu naučil. Naučil jsem se používat výše zmíněné technologie na přijatelné úrovni.

Na vlastní kůži jsem zjistil, jakou důležitou složku plní komunikace, ať už při týmové či individuální práci. Pochopil jsem, že komunikace je vždy na předním místě, protože právě díky ní se může dotýčný s daným problémem rychleji seznámit, zorientovat se v něm a obecně se přizpůsobit požadavkům firmy. Především to platí pro začínající zaměstnance firmy, kteří nemají ještě takové zkušenosti a je při nejmenším důležité, aby byli co nejrychleji zapojeni do dění firmy.

Dosažené výsledky bych zhodnotil jako uspokojivé, jelikož se mi podařilo naimplementovat z větší části všechny úkoly, které jsem dostal zadané. První projekt Třídění ocelí do skupin se chová přijatelně a proto může být označován za použitelnou aplikaci. To samé se týká dalších projektů, přičemž u posledního projektu SmallQT probíhají ještě nějaké úpravy, co se týče např. nasazení nové verze knihovny MetaCool 3.0. Jinak je aplikace již téměř u konce a připravena k použití.

9 Seznam použité literatury

- [1] *Společnost ITA*
URL: <<http://www.ita-tech.cz>>
[cit. 2013-4-18].
- [2] *LibHaru*
URL: <<http://libharu.sourceforge.net>>
[cit. 2013-4-22].
- [3] *LibHaru příklady*
URL: <<http://libharu.sourceforge.net/examples.html>>
[cit. 2013-4-25].
- [4] *Kurz MFC*
URL: <<http://farao.czweb.org/mfc1.htm>>
[cit. 2013-4-28].
- [5] Liberty, Jesse, *Sams Teach Yourself C++ in 21 Days*, 796 stran, Computer Press, 2007.
- [6] Prosis, Jeff, *Programming Windows with MFC*, 1168 stran, Computer Press, 2002.
- [7] Sutter, Herb, Alexandrescu, Andrei, *C++ Coding Standards: 101 Rules, Guidelines, and Best Practices*, 232 stran, Zoner Press, 2005.
- [8] Prokop, Jiří, *Algoritmy v jazyku C a C++*, 160 stran, Grada Publishing, 2009.
- [9] ITA, *Knihovna GphCtls32: programátorská příručka*, 16 stran.
- [10] *Lineární interpolace*
URL: <http://cs.wikipedia.org/wiki/Lineární_interpolace>
[cit. 2013-4-28].
- [11] *HTML, CSS*
URL: <<http://www.jakpsatweb.cz>>
[cit. 2013-4-27].

A Příloha

V této části jsou umístěny obrázky, které by svou velikostí rušily text této bakalářské práce.

ID	C		Mn		Si		Cr		Ni		Mo		V		D(act)	DCEQ(act)
	min	max	min	max	min	max	min	max	min	max	min	max	min	max		
101	0.001	0.06	0.05	0.7	0.02	0.3	0	0.1	0	0.1	0	0.05	0	0.05	4	
102	0.06	0.2	0.2	1.5	0.05	1.5	0	0.4	0	0.35	0	0.25	0	0.001	2	
103	0.2	0.6	0.2	1.5	0.05	1.5	0	0.4	0	0.35	0	0.25	0	0.1	2	
104	0.6	1.2	0.3	1.5	0.05	1.5	0	0.4	0	0.35	0	0.25	0	0.7	2	
105	1	1.2	0.3	1.5	0.05	1	0	0.4	0	0.35	0	0.25	0	0.7	2	
110	0.1	0.6	1.5	2	0.05	1.5	0	2	0	0.3	0	0.15	0	0.25	2	
111	0.4	0.7	0.2	2	1	2	0	0.8	0	0.1	0	0.25	0	0.001	2	
137	0.4	0.7	0.2	2	1	2	0	1.5	0	0.1	0	0.25	0	0.001	2	
140	0.1	0.6	0.4	1	0.05	0.4	0	2	0	4.5	0	0.1	0	0.1	2	
141	0.1	0.6	0.4	1	0.05	0.4	0	2	0	4.5	0	1.2	0	0.1	2	
132	0.6	1.2	0.2	1.2	0.05	0.4	0.4	3.55	0	0.3	0	0.7	0	0.7	1.5	
120	0.15	0.6	0.4	1	0.05	0.4	0	0.4	0	0.6	0.2	1.5	0	0.1	1.5	
130	0.15	0.6	0.2	1.2	0.05	0.4	0.35	3.55	0	0.3	0	0.7	0	0.4	1.5	
131	0.15	0.6	0.2	1.2	0.05	0.4	3.5	7	0	0.3	0	0.7	0	0.4	1.5	
136	0.6	0.8	0.4	1	0.05	0.4	1	5.5	0	0.3	0.2	1.5	0	1	1	
133	0.15	0.6	0.4	1	0.05	0.4	0.4	1.5	0	0.3	0.2	1.5	0	1	1	
134	0.6	0.8	0.4	1	0.05	0.4	0.4	1.5	0	0.3	0.2	1.5	0	1	1	
135	0.15	0.6	0.4	1	0.05	0.4	1	5.5	0	0.3	0.2	1.5	0	1	1	

Nastavení hodnot

Algoritmy ke třídění
Algoritmus 2

CEQ aktuální oceli
0

k
0.01

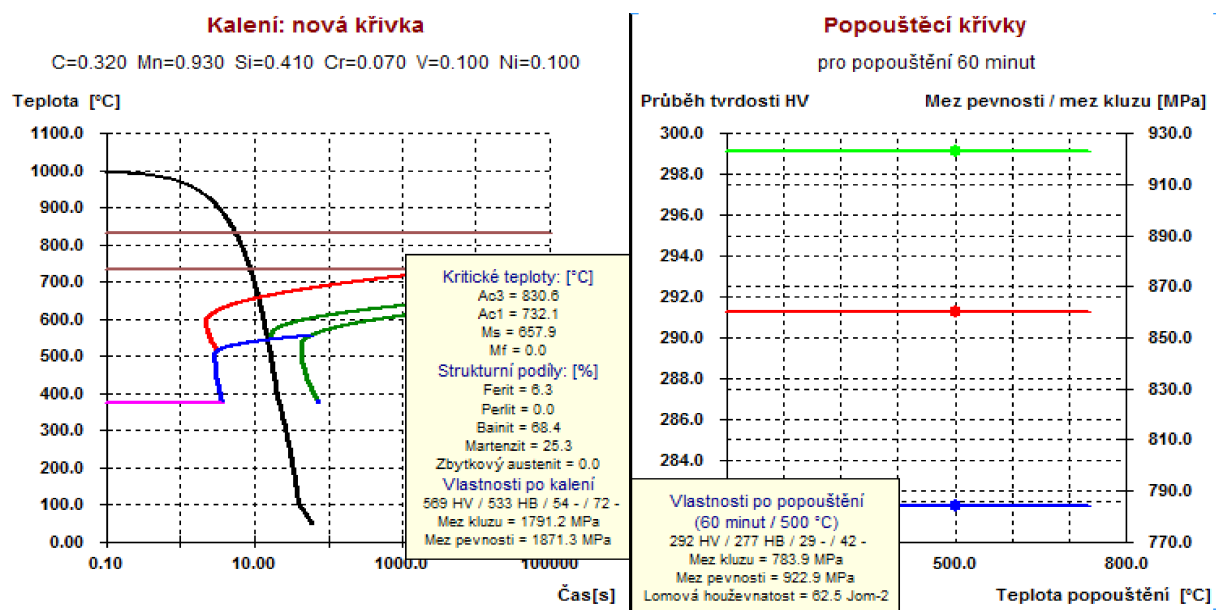
MINUS
0

Chemické složení oceli

0.01 0.1 0 0 0 0 0
C Mn Si Cr Ni Mo V

Setřít (L1) Editovat
Setřít (L2) Zpět
Výchozí
O programu Konec

Obrázek č. 2: Aplikace Třídění ocelí do skupin



Obrázek č. 3: Výstupní dialogové okno po provedeném výpočtu u aplikace SmallQT

QTSteel

Zpráva pro Projekt: **podivné vrstevnice MPROP**

Struktura a mechanické vlastnosti pro jednu ochlazovací křivku

Vlastnosti oceli:

Skupina ocelí: **17**

Chemické složení:

Prvek	Zastoupení v [%]	Prvek	Zastoupení v [%]	Prvek	Zastoupení v [%]
C:	0,220	W:	0,000	Mo:	0,000
Mn:	0,530	V:	0,000	N:	0,000
Si:	0,220	Ti:	0,000	B:	0,000
Cr:	0,000	Nb:	0,000	Ni:	0,000
Al:	0,000	Sn:	0,000	Cu:	0,000
S:	0,000	P:	0,000		

Vlastnosti počáteční austenitické struktury:

Velikost zrna: 30,000 [um]

Teplota austenitizace: 1000,000 [°C]

Obrázek č. 4: Ukázka reportu ve formátu HTML před konverzí

QTSteel

Zpráva pro Projekt: **podivné vrstevnice MPROP**

Struktura a mechanické vlastnosti pro jednu ochlazovací křivku

Vlastnosti oceli:

Skupina ocelí: 17

Chemické složení:

Prvek	Zastoupení v [%]	Prvek	Zastoupení v [%]	Prvek	Zastoupení v [%]
C:	0,220	W:	0,000	Mo:	0,000
Mn:	0,530	V:	0,000	N:	0,000
Si:	0,220	Ti:	0,000	B:	0,000
Cr:	0,000	Nb:	0,000	Ni:	0,000
Al:	0,000	Sn:	0,000	Cu:	0,000
S:	0,000	P:	0,000		

Vlastnosti počáteční austenitické struktury:

Velikost zrna: 30,000 [um]

Teplota austenitizace: 1000,000 [°C]

Obrázek č. 5: Ukázka reportu ve formátu PDF po konverzi